View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Publications at Bielefeld University

Entwicklung eines integrierten Programmsystems zur Unterstützung bei Warentermingeschäften in APL

Peter Naeve, Bielefeld

1 Einleitung

Es wird nicht ein Anwenderbericht der Art gegeben, daß über eine bestimmte Problemlösung vorgetragen wird unter dem Motto: "how I did it!"

Es wird berichtet werden über Dinge, die man beim Lösen eines konkreten Problems sozusagen problemüberschreitend gelernt hat. Motto: "how one should solve it!"

Aber es wirkt vielleicht überzeugender, wenn man zuerst einmal deutlich macht, daß man sich tatsächlich mit einem Problem – das jetzt als Basis allgemeinerer Erkenntnisse dient – beschäftigt und es vielleicht sogar gelöst hat.

1.1 Das Problem

Wie kann man einen leitenden Angestellten in seiner Arbeit durch "EDV" unterstützen? Ein Segment seiner Aufgaben – oben beschrieben als Warentermingeschäft – ist gekennzeichnet durch:

- Termingeschäfte (nur Einkauf) mit einem wichtigen Rohstoff bei nur wenigen Anbietern
- Kaufstrategien verbergen
- Rohstoff muß für die Produktion kontinuierlich vorhanden sein
- Preis abhängig von ausländischen Währungen
- Preis abhängig von LME (London metal exchange)
- Viele "insider" - Informationen
- Poolmanager (versucht Preis zu beeinflussen).

1.2 Der Rahmen

Das Projekt wurde in Kooperation zwischen der Universität Bielefeld und der Firma Schering (Bergkamen) abgewickelt. Die vorzeigbaren Ergebnisse (Programme und Paper) wurden von den drei Studenten Susanne Dahms, Christoph von Basum und Stefan Becker erstellt. Ideen, Kritik und Anregungen wurden von den Herren Jäger und Hanebeck (Schering) sowie Steinecker, Götte und Naeve (Uni Bielefeld) beigesteuert.

1.3 Die Lösung

Ein EDV-gestützter Arbeitsplatz mit:

- Graphik als Werkzeug und Sprache
- Analyse der Käufe
- Prognose von Kursen
- PC-XT (640KB, Monochrom- und Farbbildschirm)
- APL

1.4 Erfahrungen

Aus der Fülle der Erfahrungen seien vier besonders wichtige herausgestellt:

- Die "falsche" Lösung des "falschen" Problems!
- Interaktive Graphik!
- Entwicklungs- und Programmierumgebung für Neu- und Umdenker!
- Hilfe vom elektronischen Tutor!

Diese Punkte dienen zur Gliederung dieses Papiers. Wenden wir uns ihnen im Detail zu.

2 Die "falsche" Lösung des "falschen" Problems

Unklar ist, ob mit den folgenden Ausführungen eine Antwort auf Dijkstras Frage: "how do we tell truths that might hurt?" gefunden wurde. Unbestritten aber geht es hier um eine der nicht gern gehörten Wahrheiten. In diesem Abschnitt sind die Erfahrungen und Beobachtungen einer Vielzahl eigener und fremder Projekte eingegangen. Die "neue" Erfahrung in dem obigen Projekt liegt in der Erkenntnis, daß der in der Projektarbeit eingeschlagene Weg erfolgversprechend ist.

Dauernd werden Probleme gelöst, offen bleibt, ob es die richtigen Probleme sind, die man löst, und ob die angebotene Lösung auch die gesuchte Lösung ist. Dies klingt paradox! – Insbesondere wenn man die eher euphorisch klingenden, immer positiv gestimmten veröffentlichten Berichte liest. Aber es gibt für den, der unvoreingenommen die Dinge betrachtet, viele empirische Belege für die Realität der in der Überschrift angedeuteten Gefahr.

Fest steht, jemand hat das Gefühl, irgend etwas sei nicht in Ordnung - er hat ein Problem. Bald gibt es auch einen Versuch einer "inhaltlichen" Bezeichnung für das Problem, der nur allzu schnell zur definitiven Beschreibung des Problems umgewandelt wird. Macht man sich die Mühe des unbeeinflußten Nachdenkens, dann stellt man nur allzu oft fest, daß das Problem ganz woanders angesiedelt ist. Alle Projekte, mit denen ich in der letzten Zeit beschäftigt war, zeichneten sich durch eine derartige Fehlproblemstellung aus.

Selbst wenn es gelingt, durch eine sorgfältige Systemanalyse das richtige Problem zu identifizieren, so ist noch lange nicht jede Lösung eine angenommene Lösung. Akzeptanz ist die hier zutreffende allgemeine Beschreibung der Situation. In der gegenwärtigen Diskussion im Software Engineering wird unsere Sachlage auch durch die sehr philosophisch klingende Aussage beschrieben, das "Werkzeug" muß in den "Horizont" des Benutzers passen, dazu ist es notwendig, daß Probleminhaber und Problemlöser ihre "Horizonte" angleichen. Nur dann wird akzeptiert, daß es wirklich dieses Problem ist, das man hat, und daß es diese Lösung ist, die man haben wollte.

Wie kann dieses nun in der Praxis erreicht werden? Gemeinsames Analysieren der Problemlage ohne rigiden Zeittakt und rigiden Schrittkatalog ist die Antwort. Alles was der Probleminhaber sagt, kann von Bedeutung sein. Schlagwortartig könnte man sprechen von:

- benutzerorientierter Systemanalyse
- benutzergetriebener Systemanalyse.

Den Prozeß der Lösungsfindung sollte man gestalten wie die Produktion eines Herrenanzuges beim Maßschneider in der guten alten Zeit. Da konnte man bei den Anproben durchaus mehr als nur das "paßt noch nicht" beheben. Es war auch möglich, neue Formvarianten oder modische Trends nachträglich zu berücksichtigen. Auch hier könnte man schlagwortartig zusammenfassen zu

- benutzerorientierter Problemlösungsprozeß
- benutzergetriebener Problemlösungsprozeß

Ein in der Praxis gern gebrauchtes Wort der Entschuldigung lautet: "das Tagesgeschäft!" Das Tagesgeschäft hat den Zeitplan fest im Griff. Es verhindert, in einiger Muße über die Dinge nachzudenken, vermeintliche Umwege zu gehen usw.. Im hier geschilderten Projekt gelang es, einen Arbeitsstil zu etablieren, der geeignet scheint, vorstehende Schwierigkeiten zu vermeiden. Ausführliche Problemanalyse mit den zukünftigen Benutzern halfen allen Beteiligten, einen "gemeinsamen Horizont" zu entwickeln. Die Lösung wurde durch häufige zeitrestriktionsfreie "Anproben" davor bewahrt, nur die Lösung des "Problemlösers" zu werden. Dabei wurde mehr als einmal auch die Problemstellung neuen Erfordernissen angepaßt.

3 Interaktive Graphik

Es ist schon so, daß man es gar nicht mehr hören mag, das Wort "Graphik". Eine wahre Flut von "Graphiken" – kumulierend in dem schrecklichen Begriff "business graphics" – ist über uns hereingebrochen. Und doch, es lohnt, sich noch einmal mit "Graphik" zu beschäftigen.

Unser Umgang mit den mächtigen Möglichkeiten, die uns die graphischen Fähigkeiten von Hard- und Software schon jetzt bieten, scheint uns von wenig Phantasie geprägt zu sein. Phantasie aber wird uns helfen, Dinge zu erträumen, die zwar vielleicht jetzt noch nicht realisierbar sind, deren Realisierung wir dann aber kraftvoll fordern sollten. Warum immer mit dem zufrieden sein, was man bekommt?

Damit wir uns ganz ohne technische Ablenkungen und Restriktionen unseren kreativen Wünschen hingeben können, sind die folgenden Folien bewußt nicht mit Hilfe des Computers erstellt worden. Erst die Vision, dann die Implementation heißt die Botschaft.

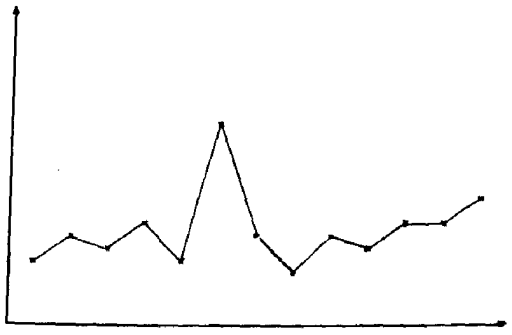

Abbildung 1

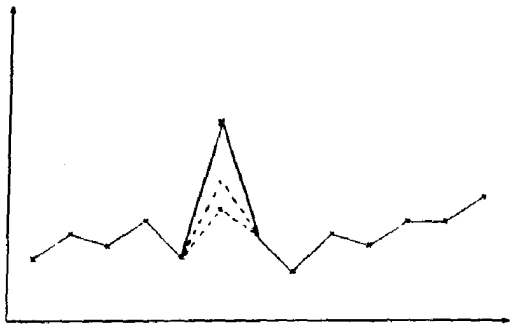

Abbildung 2

Die *Abbildung 1* zeigt den Graphen einer Zeitreihe (Rohdaten). Bekannt sei,daß einige spezielle Ereignisse in der Vergangenheit dazu führten, daß die entsprechenden Zeitreihenwerte nicht "typisch" sind, d.h. bei Prognosen z.B. nicht mit vollem Gewicht eingehen sollten. Wünschenswert wäre die Möglichkeit, auf die entsprechenden Zeitreihendaten zu zeigen und die Werte zu korrigieren. Dies alles in der Graphik, wobei die numerischen Werte zur weiteren Bearbeitung an die entsprechenden Programme weitergegeben werden.

Die *Abbildung 2* verdeutlicht einen ersten Wunsch. Er ist gekennzeichnet durch:

- Herausgreifen eines Punktes
- Festlegung einer Umgebung
- Verschiebung des Punktes unter Veränderung der Umgebung.

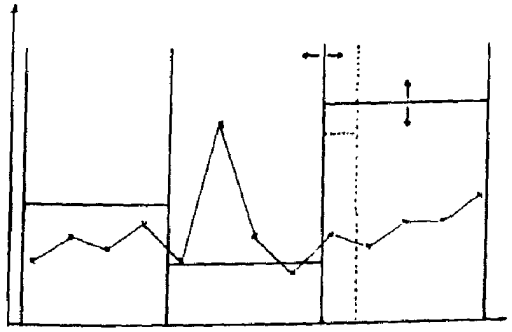

Abbildung 3

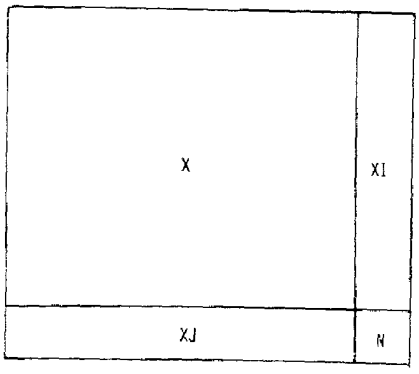

Abbildung 4

Skizzierten wir eben die Behandlung (Filterung) isolierter Punkte, so wenden wir uns jetzt der globalen Gewichtung zu. Man möchte unterschiedliche Zeitperioden mit unterschiedlicher Gewichtung in nachfolgende Berechnungen eingehen lassen. Die *Abbildung 3* zeigt eine andere Wunschvorstellung. Die wesentlichen Handlungen, die man ausführen können möchte, sind:

- Festlegung von Bereichen gleichen Gewichts
- Veränderung dieser Bereiche
- Festlegung des Gewichtes in den Bereichen
- Veränderung dieser Gewichte
- Bereitstellung der numerischen Werte der Gewichte für weiter Rechnungen

Was wird benötigt, wenn wir derartige interaktive Graphiken "bauen" wollen?
Sicher nicht die üblichen "canned" Produkte!
Ehe die interaktive Graphik unseren Vorstellungen entspricht, findet ein längerer Prozeß des Experimentierens und Revidierens statt. Dies muß die Software zulassen. Vorgriff: APL eine Umgebung zum "rewriting and rethinking" erweitert um

- Schichtenkonzept
- AP333 bringt APL Basisfähigkeiten in Graphik bei - konnte damals von IBM nicht bereitgestellt werden
- Ein Satz von Grundfunktionen als Bausteine für höhere Graphiken, z.B. Skalierung, Klippen, Achsen, usw.

Auf dieser Basis setzt der kreative Entwickler auf. So ist es in dem Projekt GAP mit Erfolg praktiziert worden. Es wurden "spielerisch" viele Möglichkeiten ent- und verworfen, experimentell erprobt und "on the job" modifiziert.

4 Entwicklungs- und Programmierumgebung für Neu- und Umdenker

Die Ausführungen der beiden vorstehenden Abschnitte haben deutlich gemacht, daß besondere Anforderungen an die Entwicklungs- und Programmierumgebung gestellt werden. Wenn man das Konzept des fortwährenden Revidierens und Experimentierens in die Tat umsetzen will, benötigt man eine Umgebung ohne großen programmtechnischen "Overhead" für jede einzelne Maßnahme. Es war schon der Name des Kandidaten unserer Wahl gefallen: **APL**.

Darauf wird es wie üblich zwei Reaktionen geben. Eine kleine Gemeinde von APL-Kennern wird mit leuchten Augen zustimmend nicken. - Sie können den Rest dieses Abschnittes überspringen. - Die Mehrheitsmeinung lautet schlicht: APL ist nur für mathematisch angehauchte Spinner, aber nicht für ernsthafte kommerzielle Anwendungen geeignet. Eine Meinung umso bestimmter vorgetragen, je weniger Kenntnisse von und über APL vorhanden sind.

Leider kann keine Demonstration von APL und seinen Möglichkeiten wegen der begrenzten Seitenzahl hier präsentiert werden. Es seien aber einige allgemeine Punkte angeführt, die zeigen, daß APL für die Umsetzung des oben erwähnten Konzeptes gute Voraussetzungen mitbringt:

- APL ist ein interaktives System (Sprache)
- APL ist eine interpretative Sprache
- APL hat mächtige Operatoren (Funktionen)
- APL hat mehrdimensionale Datenstrukturen

Die ersten beiden Eigenschaften erleichtern gerade den Prozeß des Revidierens und Experimentierens. Die beiden letzten Eigenschaften erlauben eine kompakte Programmierung, da man auf einer viel höheren Ebene ansetzen kann als bei den "eindimensionalen" Sprachen wie z.B. PASCAL.

Der Versuchung ein kleines Beispiel zu geben, kann ein APL-Missionar nicht widerstehen. Es zeigt, wie gut APL mit Tabellen umgehen kann. Unbestritten spielen Tabellen eine große Rolle in Anwendungen. Die hier betrachtet statistische Problemstellung ist von mehr als nur akademischem Interesse. Schaut man hinter die Kulissen von unserem Programmsystem GAP, so erblickt man auch dort Tabellenmanipulationen ohne Zahl. Die wenigen Abbildungen müssen als kleiner Einblick genügen. Sie zeigen die Stuktur des Problems (*Abb.4*), die APL-Lösung (*Abb.5*) und eine PASCAL-Lösung für ein Teilproblem (*Abb.6*).

```
X  observed cell frequencies
N  number of observations
XI row margins (row totals)
XJ column margins (column totals)
XI ← + / X      ⍝OR⍝  XI ← + / [2] X
XJ ← + ⌿ X      ⍝OR⍝  XJ ← + / [1] X
N ← + / + / X   ⍝OR⍝  N ← + / + ⌿ X
⍝ next line calculates augmented contingency table ⍝
XC ← XX ,[1] + ⌿ XX ← X, + / X
```

Abbildung 5

```
for j := 1 to n do
begin
xj[j] := 0 ;
for i := 1 to m do
    xj[j] := xj[j] + x[i,j] ;
end
```

Abbildung 6

5 Hilfe vom elektronischen Tutor

Auch bei noch so benutzerorientiertem Vorgehehen läßt es sich nicht ganz vermeiden, daß zwischen Benutzer und Programmsystem eine Restfremdheit bestehen bleibt. Dafür gibt es eine Vielzahl von Gründen:

- Das Programm ist in der Regel in seinen Möglichkeiten komplexer als der Benutzer auf Grund der Anproben vermutet. Ein gutes Programm transzendiert seine ursprünglichen Ziele. Dies hängt ganz wesentlich von der Kreativität des Entwicklers ab und inwieweit er selber während der Systemanalyse das Problem zu seinem machen konnte.
- Der Kreis der Benutzer ist nicht fix. Neue Benutzer müssen an das Programm herangeführt werden.
- Da der Umgang mit dem Programm eine von vielen Aufgaben des Benutzers ist, kann durchaus ein technischer Handgriff momentan nicht völlig gegenwärtig sein

Natürlich sind dies keine neuen Feststellungen. Und es gibt schon eine ganze Reihe Vorschläge, wie man den Benutzer in solchen Fällen unterstützen sollte. Um nur einige zu nennen:

- Handbuch
- HELP-Feature
- Sample-Programm

Allen diesen Vorschlägen sind zwei grundlegende Mängel gemeinsam. Zum einen lösen sie nur jeweils eine der Schwierigkeiten auf und zum anderen "helfen" sie dem Benutzer nicht direkt in seinem Arbeitszusammenhang. Sie nehmen ihn immer aus seiner Arbeit heraus. Außerdem hat man höchst selten den Eindruck, daß sich die Realisationen obiger Hilfen an Lehr- und Lerntheorien orientieren.

In unserem Projekt wurde versucht, den kritischen Anmerkungen gerecht zu werden. Die gefundene Antwort heißt "Computer Aided Tutorial". Dahinter verbirgt sich eine Kombination zweier häufig diskutierter Vorgehensweisen.

- *Branching Tutorial* ist der Versuch, den Benutzer bei der Gestaltung des Kursaablaufes zu beteiligen.
- *Simulation* ist eine Antwort auf die Frage, wie Wissen in Handlung umgesetzt werden kann. Dem Benutzer wird in Situationen ermöglicht, die Reaktionen auf die Handlungen, die er aus seinem Wissen ableitete, zu studieren, um so sein Wissen zu verstärken oder zu revidieren.

Die gefundene Lösung lässt sich ohne Rechner nicht demonstrieren. Man muß den Tutor in "action" sehen. Nur so viel sei noch gesagt, der Tutor ist immer da. Wenn der Benutzer es will, beginnt er mit seiner Hilfe in der jeweiligen Arbeitssituation. Er leitet den Benutzer bis dieser meint, alleine weiter vorgehen zu können. Dabei wird der Benutzer nie in einen anderen Arbeitszusammenhang geführt. Er macht die ganze Zeit seine Arbeit – auf Wunsch vom Tutor unterstützt.

6 Schluß

Zusammenfassend sei die Frage beantwortet, was war die Botschaft?

- Muße hat Wert!
- Phantasie ist erlaubt!
- Auch fordern, nicht nur konsumieren!
- **APL** ist gut!

7 Literatur

v. Basum Ch. Entwicklung eines integrierten Programmsystems zur Unterstützung bei Warentermingeschäften in APL
– Datenorganisation und Kaufanalyse –
Diplomarbeit Fak. f. Wirtschatswissenschaften Uni Bielefeld
1986 unveröffentlicht

Becker St. Entwicklung eines integrierten Programmsystems zur Unterstützung bei Warentermingeschäften in APL
– Grafische Darstellung und Prognosesystem –
Diplomarbeit Fak. f. Wirtschatswissenschaften Uni Bielefeld
1986 unveröffentlicht

Dahms S. CAT – Computer Aided Tutorial
Konzeplt und Entwicklung eines offenen Tutorsystems
Am Beispiel eines Programms zur Unterstützung bei Warentermingeschäften
Diplomarbeit Fak. f. Wirtschaftswissenschaften Uni Bielefeld
1987 unveröffentlicht

Dijkstra E. W. Selected Writings on Computing:
A Personal Perspective
1982 Springer Verlag, New York

Winograd T., Flores F. Understanding Computers and Cognition
A new Foundation for Design
1986 Ablex Pub., Norwood